



MAY 1985

CONTENTS

2. Information on the Group
3. Editorial
4. Members' letters
7. Problem section
10. QL crashes when using CALL
11. Disk sector read/write
12. Disk firmware upgrades
12. Software and hardware update
13. Errata
14. Eidersoft QLArt
14. Using memory expansion and disk together
15. Review of QC compiler
17. Simplex RAM expansion - an update
18. Abacus
18. A file dump programme in BCPL
20. Olivetti Praxis 35
21. Begin again
22. Retractablebug
23. Book review - 'An Introduction to Programming the Sinclair QL' by R.A. and J.W. Penfold
23. Book review - "Understanding the 68000" by Leon Heller
24. Some notes on Archive
25. Printing from QLChess
26. 68000 MOVE instruction codes and QL version JS
27. Computer art!
28. Example programmes for the 68K/ASM assembler
29. Serial numbers etc.
30. Aylesbury QL user Group
31. Using QLChess with additional RAM
31. Merge overlay technique
33. Archive bug
34. Window initialisation of library programmes
35. Psion upgrades
36. Machine code extensions to read current graphics coordinates and turtle heading
38. Book review - "Sinclair QL User Guide" by Lionel Fleetwood
39. Secretary's notes.

INFORMATION ON THE GROUP

Membership of IQLUG is by subscription to Quanta, the group's newsletter, which is published monthly. Membership details are obtainable from the secretary. Membership of the group is open to anyone with an interest in the Sinclair QL microcomputer.

Members requiring assistance with problems related to the QL may write to or 'phone the secretary. An attempt will be made to put them in touch with a member who can help with the problem.

Workshops will be arranged from time to time in various parts of the country. The group maintains a software library. Most of the programmes are free to members. A library list will be published from time to time. programmes are obtainable from the sub-librarians.

A list of members in any particular area is obtainable from the secretary.

Please send all contributions to the newsletter to the editor.

Chairman and
Newsletter Editor

Leon Heller,
8 Morris Walk,
Newport Pagnell,
Bucks. MK16 8QD.
Tel: (0908) 613004

Secretary and
Newsletter Publisher

Brian Pain,
24 Oxford Street,
Stony Stratford,
Milton Keynes MK11 1JU.
Tel: (0908) 564271

Sinclair, QL, QDOS, QLUB and ZX Microdrive are trade marks of Sinclair Research Ltd.

Quill, Abacus, Archive and Easel are trade marks of Psion Ltd.

NOTES FOR CONTRIBUTORS TO QUANTA

Contributions should be printed on A4 paper, using a fresh ribbon. Do not fold the sheets, but keep them flat when posting.

Quill files on Microdrive cartridges (cartridges will be returned) are acceptable, as are handwritten articles

Short programmes can be included, but lengthy programmes should be placed in the library, and a description sent in for the newsletter.

FOR SALE

GST 68K/OS Operating system with Utilities	Cartridge Assembler.
Programmer's Manual.	£95
COMPUTER ONE PASCAL	£25
METACOMCO Assembler (new version)	£25
PSION V2 Software (with manual updates)	£18
QCODE Terminal emulator Assembler	£12
RODENT QL Artist	£5
Miracle Systems Centronics Interface	£18
Adder Pub. QL Advanced User Guide(Dickens)	£9
SPECIAL OFFER ALL the above	180 pounds.

Contact Wilson Stothers 041-339-3688

EDITORIAL

We are sorry about the delay in getting the April issue out. The printer took longer than usual and we had a two week postal strike in Milton Keynes. This issue should be on time. Unlike the commercial magazines, Quanta should go out during the month on the cover.

We are finding it very difficult to get adequate supplies of cartridges, so that all the software sub-librarians can have copies of the latest library software. I managed to locate 25, which have been sent to Alan Bertram for distribution, and hope to have located some more by the time you receive this issue. This shortage of cartridges is what has been causing the inordinate delay in fulfilling requests for the software library. We hope to resolve the problem soon.

Partly because of all the hassle over the lack of cartridges, and also because of the tone of a letter Alan received from one member over a sub-librarian's inability to fulfil a request for the latest software, Alan has resigned from the post of Software Librarian. Sid Day has very kindly agreed to take the job on. Sid's address is:-

261 Highfield Road,
Romford,
Essex R75 3AW.
Tel: (0708) 27272

In "Computing The Magazine", Sinclair are quoted as saying that the return rate for QLs is about 3% to 4%. This is excellent, if the figures are to be believed.

Sinclair claim to have sold 60,000 QLs to date, which isn't much as they planned to sell 100,000 last year. They are hoping to sell 200,000 by the end of this year, but I get the impression that it is currently doing rather badly, perhaps because people that might be interested in an advanced home/small business computer are waiting for the ATARI ST machines to arrive. Both GST and Metacomco tell me that they will be putting their software on the "Jackintosh", as will several other software houses. However, it looks like the early ST machines will be in a similar state to the early QLs, with unfinished and bug-ridden software, and fully functional machines might not be available until the Autumn, which gives Sinclair some time to get their act together.

According to a local paper, a denizen of Bletchley, Milton Keynes, with a team of seven colleagues, intends to drive/pedal a Sinclair C5 trike from John O'Groats to Lands End, in aid of charity!

Leon Heller

MEMBERS' LETTERS

Having received a number of back copies of QUANTA recently, I was really delighted with the content and am looking forward to further editions, perhaps making a contribution to the forum myself. My interests in computing mainly rest in the area of educational uses of computers, both as visual aids and in teaching computer familiarity to newcomers. This is a most rewarding aspect of computing.

I have written a programme in BASIC, on Research Methodology. It is written for both the teacher as a classroom aid, and for the individual student who is interested in the subject. If my enthusiasm continues I would hope to write a companion programme on statistics which will link in to the research programme.

A friend and I are arranging a weeks course in BASIC programming for beginners. We have got very favourable rates in a local hotel and, what with computing, the indoor swimming pool, squash court and sauna we hope to sweat off a few pounds and have some fun in the process. If any 'IQLUG' members are interested they may like to telephone or write to me as I am sure that they will find the rates, especially to IQLUG members, very attractive.

Gerry Meenz,
13, Westminster Drive,
Haydock,
Merseyside WA11 0RU.

Does everyone realise that a standard 180 degree 5 pin DIN plug will supply all of the signals required for a composite video monitor - there's no need to look for 8 pin plugs.

NOTE in SuperBASIC reverses the truth of the following statement, which is often very useful:-

```
IF NOT INKEY$(-1) = 'Y': EXIT
```

One of my Microdrive cartridges recently stopped, with the tape jammed up inside. So I opened it up, untied the knot, rethreaded it, and got back my file. Having proved this is possible I would strongly recommend that anyone with a similar problem throw the cartridge away quickly before they damage their mental health. For those as daft as I am, count the turns as they fall off and put the same number back. Don't worry about loops two feet long hanging out - they will go back, eventually.

In praise of an RS-232 that works. I recently connected the QL to an ancient three cwt. of ancient Diablo printer and dumped a file to an Apple and

received both a text file and an object file all in the same day, and without a single hardware mod. I believe this could be a record.

I also use the CP/M User Group 68000 cross-assembler, and I think there's a bug in MOVEP - it leaves out the displacement.

As the QL doesn't have a screen clear command (only window clear) Sinclair use the following code:-

```
        MOVE.L      £131072,A1
        MOVEQ      £0,D6
LOOP:   MOVE.L      D6,(A1)+
        MOVE.L      D6,(A1)+
        .....     TOTAL EIGHT TIMES
        MOVE.W      A1,D1
        BPL        LOOP
        RTS
```

My FIG-FORTH 2.1 is up and running, but without storage (see problem section) I won't send it to the library yet. Anyone wanting a copy send a cartridge.

Electronics and Computing Nov. '84 had a 'directory read M/C utility that looks like the basis of some better copy utilities and all the other things we need a usable dir for. I feel the best function would be one that reads the directory into a BASIC array, called say, DIR\$(file no.). DIR(0) would contain the number of files, and possibly the media name. DIR(1) onwards would be the 64 byte headers from the file, so the 'difficult' concept of string slicing can find anything you want to know about all the files on the drive. Does this fill a long felt want? Does the format look like a good standard? Does anyone want to write it? Well, I might soon if pushed.

David Stevenson,
PO Box 508,
Muscat.

/* A three pin plug may be used if you only want monochrome output.

The QC compiler comes with a function called "readdir" that does most of the above. */

My experiences parallel those of many readers, but at a longer distance from the UK. Sinclair would not ship out of the UK because of a local dealership contract. They also stated that the QL would not work here because of power and TV channel differences. Having ordered through my UK address I was treated to the usual lies about delivery dates, and finally got the QL last September. I can confirm that Sinclair's box fits into the overhead luggage bins

on 747s, but not under seats. You can buy a big plastic carrier bag at Heathrow for the box, but the handles cut into your fingers!

Of course, the QL works here; we have 240v 50Hz power. For those of us with VHF TV, the video recorder provides UHF Ch36, although most new TVs have UHF/VHF tuners. Yes, the software is terribly slow and TV resolution poor; a Kaga Vision II monitor gives perfect resolution but why make the QL output 85 columns when 80 is an industry standard?

I ordered a Q Pi interface from CST for airmail delivery in November '84 and paid handsomely. It arrived 12 weeks later by surface mail. Despite exports being VAT-free, I did not get a refund and still wait. The price in the UK dropped to Less than £50 at about the time mine was dispatched, -as a result of this one might be forgiven for doubting the integrity of the supplier.

I also ordered Typing Tutor from Computer One. Once again the item was surface posted and the p and p pocketed and not refunded. The actual software is poor, the errors being a combination of poor on-screen layout, incorrect spelling and references to keys which do not exist / features which do not exist. To cap it all, I can manage about 50 wpm on programme startup. The programme acknowledges this speed and says it can improve my rate to 30 wpm.

From CST I have requested a refund. From Computer One a new version or a refund would do., I wonder what their other products are like if I have had such trouble with such simple things?

As an Apricot dealer, and a user of Cromemco Z2 and DecMate II systems I would hate to be involved in such a scam as the QL at the present time. I have abandoned plans to import QLs for resale here and have abandoned plans to import add-ons too, as a result of my experiences, although I will still look out for new products on my next visit.

Russell Smith,
PO Box 190,
Heidelberg,
VIC. 3084,
Australia.

/*, Martin Baines of CST tells me that they were unable to send Russell's Q Pi interface airmail, because it was over the maximum weight limit. They also had to get an export licence for the item, involving a lot of extra paperwork, which cost about the same as the VAT refund would have been. CST do seem to have taken their time to respond to Russell's letters, however.

I raised Russell's criticisms of the Typing Tutor with John Scott of Computer One. He considers that Russell was overly critical, in that the problems he mentions are quite minor. One word is misspelled, and one menu item is misplaced by a character position. The incorrect speed when the programme is first used is due to a bug; the programme gets the initial typing speed from the speed of entry of the user's name, which can obviously be entered very quickly (>30 wpm), whereas the maximum speed the programme can handle is in fact 30 wpm. */

I would like to inform you of the actual situation of the QL in Belgium and France.

In Belgium, 3000 QLs have already been distributed, but actual sales are slowing down due to lack of disks and programmes.

In France, 1000 English QLs were distributed around Christmas. The French "azerty" version is further delayed from April 15 to May.

R. Betz,
President,
Club Micro-QL,
38 Chemin du Moulin,
1328 Ohain,
Belgium.

PROBLEM SECTION

I have still not found a simple command or procedure that will enable me to reproduce the screen display direct to my printer, an Epson RX-80F/T+. There must surely be a very simple instruction available, that so far I have been unable to define. The printer responds fine with Quill, LIST, etc., but I cannot make it copy the screen display!

Thank you for QUANTA, even though only about 10% of your material and members contributions are comprehensible to me. If someone can offer the solution to my small query, then maybe my learning curve will improve!

V. G. Avery,
Officers' Mess,
RAF Laarbruch,
BFPO 43.

/* I think this has been covered in an earlier issue: you use the Easel GPRINTPRT programme. */

I note that Colin Opie in his book 'QL Assembly Language' says with regard to the 21 or so arithmetic and other utilities linked with BASIC (e.g. p 162) that all addresses must be relative to A6. I can understand what is meant (A6,A1. L), but it is far from clear to me exactly how one carries this out. I find it quite annoying that some of these books exhort you to do certain things, but dodge giving a few lines of assembler as an illustration of how to do it. And again (same page), when you make A1 the pointer to the stack, do you need to make room on the stack? And if so, is this a question of adding several zero words to the stack? And how does one tidy up afterwards? I have tried to do what is suggested with signal lack of success.

Bill Cowhig,

21 Priory Road,
Sale,
Cheshire M33 2BS.

/* This is one of the many omissions in Opie's book, although it is covered in the example on page 219 - Extra BASIC Functions! Its a pity Opie didn't liaise better with Tony Tebby, who wrote the examples, and made sure that he covered the same material. Adrian Dickens' book mentions BV.CHRIX which makes room on the arithmetic stack, and the other utility vectors you need when accessing the arithmetic functions. */

I have sent off for my QL Technical Guide, but in the meantime I would appreciate help here. When I attempt to use QDOS vector BV.CHRIX to obtain space on the arithmetic stack, on return I have -1 in D0.L and status flag Z = 0 for allocations up to 31 bytes. For allocations of 32 bytes plus, I get D0 = -1 and Z = 1, but from 64 bytes onwards I get D0 = \$58 and Z = 1, and this seems to go on until 32K., but crashes BASIC when a 128K allocation is requested. Neither A6 or BV.RIP(A6) changes at all. When data were written in following a 64 byte request there is no problem, but for 128 bytes, QDOS crashes.

Ivan Krishnan,
12, Giles House,
168, Westbourne Grove,
London W11 2RJ

/* I feel you would be better off with Adrian Dickens' QL Advanced User Guide than the Technical Guide, as the former has all the information in the latter, and a lot more besides. BV.CHRIX returns with D1, D2 and D3 undefined, and an undefined error return in D0. You are certain to get a crash when trying to allocate 128K bytes on the arithmetic stack, anyway, or do you mean 12B bytes? All you need to do is:-

```
MOVEQ    £$XX,D1
MOVE.W   BV_CHRIX,A2
JSR      (A2)
```

to allocate \$XX bytes on the arithmetic stack. You only need to allocate a few bytes, as a rule, depending upon the complexity of the arithmetic function you wish to evaluate. Six bytes per number are required, I think. */

I recently wanted to move the cursor back one place on the screen, by printing a BS character. By convention, this is a different character on all machines, and a string of characters like 08, 08, 20, 20, 08, 08 to achieve a delete is not unusual. However, the QL goes one better - there is no character backspace. \$C2 is the value from the keyboard, but Sinclair only

comment that if you are writing your own backspace please use the same character. My print a character, checking for backspace, goes like this:-

```
MOVEQ      £-1,D3      TIME OUT
MOVE.L     $1001,A0    CHANNEL 1
MOVE.W     (A3)+,D1    GET CHAR.
CMPI      £194,D1
BEQ       BACKSPACE
MOVEQ     £5,D0
TRAP     £3
JSR
BACKSPACE:
MOVEQ     £$13,D0
TRAP     £3
MOVEQ     £$20,D1
MOVEQ     £5,D0
TRAP     £3
MOVEQ     £$13,D0
TRAP     £3
JSR
```

This works fine on one line, but if you have wrapped round from the line above, the move cursor back trap will report an error and not move. Has anyone any ideas on how to find where the last character on the row above is, and how to get the cursor there, in under a page of code?

Does anyone know how to use the read/write a block Microdrive vectors? My copy of Adder's Advanced User Guide is just a little sketchy on the subject. Apparently vector 124 reads a block without being told which one to read. I think pushing the file number and block number on the stack as vector 126 might help, but what about the drive number? As A7 is described as volatile do we assume it is reset to the above file number, or wholly random?

David Stevenson,
PO Box. 508,
Muscat.

Has anyone got the 'Saints' programme on page 28 of the February issue to work? It keeps crashing at line 350 with "Not found".

Gordon Darling,
PO Box 3474,

Boroko,
Papua New Guinea.

My printer is a Radio Shack Mark 8. I can get it to list etc. at 1200 baud using ser2shc, except that I sometimes lose the first character of a new line. However, with the printer plugged into ser2 and Quill/Printer installed etc., it doesn't even twitch when I say 'Print'. The software goes through all the motions, says "insert a new sheet of paper", and when I press ENTER, nothing prints out.

Does anyone have a solution which doesn't involve a new printer?

Petter Finne,
Otis Elevator Company Ltd.,
PO Box. 20014,
Nairobi,
Kenya.

QL CRASHES WHEN USING CALL

I wrote to you about a crash when making hardcopy of the screen using the standard method with GPRINT_PRT from Easel in SuperBASIC. According to technical advice from Sinclair there is "minor bug that prevents CALL from functioning as expected with SuperBASIC programmes over 32K in size". This size includes space for dimensioned arrays. The way round it is to convert any machine code you wish to use into a procedure.

Sinclair provided a programme which adds a procedure called 'dump' to SuperBASIC. It should be loaded after resetting the QL and run before running programmes having large arrays and including a screen dump or other CALLS. Note that GPRINT_PRT from Easel is in mdv1 (line 170).

Listing A

```
100  RESTORE
110    a= RESPR(3000)
120    cs=0
130    FOR i=0 TO 26 STEP 2
140    READ x: POKE_W a+i,x: cs=cs+x
150    END FOR i
160    READ ccs: IF cs<>ccs THEN PRINT 'Incorrect data - please
check': STOP
170    LBYTES mdv1_GPRINT_PRT, a+28
180    CALL a
190    DATA 17402,8,13432,272,20178,1,14,1092,21837,20480, 0,0,0,
24590, 119306
```

Here is a test programme using dump (line 160) when an array has been dimensioned to 13908 elements (lines 70, 80). This is loaded and run after the programme in listing A has been run. The REM comment in line 10 refers to the programmes "sin_dump" which is listing A and "cr4_sint" which is listing B.

```
10    REMark programme cr4_sint. Run after lrun of sin dump from Sinclair
letter    19.1-.85
40    MODE 512
70    z=13908
80    DIM x(z)
150    CLS:AT 8,8 PRINT "If hardcopy of screen printed without crash:
see      hardcopy"
160    DUMP
```

Hugh Hack,
Backus,
Shepherds Green,
Henley-on-Thames,
Oxon. RG9 4QN.

DISK SECTOR READ/WRITE

The latest versions of Tony Tebby's disk software (1.07 and above) allow one to read and write disk sectors from SuperBASIC, or any other high level language.

You first open a channel using, the file name 'flpx_*dyz' where x is the drive number ("1" or "2" with CST's interface), y is "0", "1", "2" or "3" representing 128, 256, 512 and 1024 bytes per sector respectively, and z is "s" for single-density and "d" for double-density. For QL disks the correct incantation is therefore

OPEN #3,'flp1_*d2d. for example.

Using

```
GET #3|x, a$
```

where $x = 65536 * \text{track} + 256 * \text{side} + \text{sector}$, sectors may be read into the string variable a\$. Substituting "PUT" for "GET" will result in a write to the sector. With interfaces other than CST's, you might need the Sinclair Toolkit to get the direct access feature to work. It might be possible to use Tony Tebby's SuperBASIC extensions, from the software library. If you are using C or BCPL, which allow direct (random) file access you won't need the Toolkit.

I/O must be in multiples of the sector size.

This technique obviously allows "alien" disks to be read and written to, and I have successfully accessed a single-sided single-density Model I TRS-80 disk, using a simple SuperBASIC programme. It should be feasible to transfer data from an IBM PC disk to the QL, and vice versa, using this method, and it should work with most other disk formats.

Using this facility I am writing a programme in C called QZAP that will have the following functions:-

- edit disk sectors
- transfer sectors between different disk formats
- write sectors to a file
- read sectors from a file
- fast backup (track to track) even with a single drive

and, possibly, allow other disk formats to be created, with the appropriate disk structure, so that disks acceptable to other machines may be created on the QL. The latter will be quite tricky, however. QZAP will be similar to Superzap, on the TRS-80, and the "Disk Doctor" programme for the Beeb, and will enable corrupted disks to be "repaired", or most of the important information on the disk to be put on another disk, if it is impossible to correct it.

My thanks to Martin Baines of CST, for telling me about this feature, and Tony Tebby for answering a couple of queries about it.

Leon Heller

DISK FIRMWARE UPGRADES

Martin Baines of CST has given me the following information about upgrades to their disk interface software, which is contained in an EPROM on the circuit board.

EPROMs will be exchanged free if you find a bug in any version.

Versions up to 1.06 will be upgraded free if you return the EPROM or interface to CST.

Versions beyond 1.06 will be upgraded for a £5 handling charge if you return the EPROM or interface. If you want to keep the old EPROM the charge will be £25. This option is for people who cannot do without their disks for a few days.

CST's address is Cambridge Systems Technology, 30 Regent St., Cambridge.

Leon Heller

SOFTWARE AND HARDWARE UPDATE

A lot of new software and hardware is now available, or will shortly become available, so a summary might be of interest.

TDI reckon that their implementation of the UCSD P-system should be available shortly, with UCSD Pascal and FORTRAN-77.

Metacomco have just sent me a beta test copy of their ISO Pascal. It is quite a large piece of software, and comes on two cartridges and an EPROM that plugs into the

ROM slot at the back of the machine. Memory is a bit tight, and the compiler uses some of the video RAM when it is running. They hope to have it generally available in June. Lattice. C should be ready in July.

Simplex Data have redesigned their memory expansion units. The PCB has been tidied up, the regulator now has a heat sink, and the card has a CST black plastic case. I should shortly have one of their bus expansion modules, which enables one to have both their memory expansion module, and a peripheral such as a disk interface, connected at the same time. The cost will be about £50, which includes a power supply.

Cumana will soon be supplying their range of disk drives for the QL. They will be identical to their Beeb ones, but painted black. They are putting the 68000 version of the OS-9 operating system on the QL. OS-9 is quite "Unix-like", and the product will be launched at the Unix show in June. It will come bundled with BASIC, assembler and a C compiler, as well as business packages like Dynacalc, a spreadsheet programme.

GST hope to have their disk-based 68K/OS running soon. They also intend to make their very nice 68K/OS text editor and word processor available under QDOS, and are working on an "all-singing, all-dancing" macro-assembler for both 68K/OS and QDOS

PCML have put their Z-80 add-on for the QL on hold for the time being, but they are shipping their RAM expansion units, which range from £75 for the 64K version up to £270 for the 512K model. For £10 extra you can have an EPROM socket included, allowing a 16K EPROM to be plugged in. They will be providing an EPROM programming service. They hope to have RAM disk software available in June, as well as a bus extender combined with RAM and disk.

Kempston, well known for their joysticks, are also working on a disk interface for the QL, and a parallel printer port like the CST unit.

Tandata have taken on the OEL modem for the QL. I don't know yet what their policy will be towards people who have already paid OEL for equipment. I will be contacting them on this matter.

Miracle, who produce the serial/parallel converter used by many members, are developing a mouse. They can also supply a screen dump programme, and serial leads.

Leon Heller

ERRATA

My recent contribution to QUANTA, entitled "Creating DATA statements", contained a routine that does not work correctly, due to an omission. If the following line is added, all should be well: 165 start=start+1

My apologies if that omission causes any problems.

David Nowotnik,
12 Long Plough,
Aston Clinton,
Aylesbury,
Bucks. HP22 5HA.

EIDERSOFT QLART

During my visit to the Alexandra Palace show I made the mistake of purchasing the Eidersoft QLArt software package. What a useless thing that is! It must have been concocted just with people in mind that want to see something on the screen, because that is the only thing about it that works. As soon as one attempts to produce a hardcopy (h - command) the thing crashes so it is totally useless. File copies that are produced with it are not accepted by the standard software and cannot therefore be loaded. The brochure that came with it falls into the same category - it is absolute rubbish! When I had finished reading the first two or three trade magazines I was more articulate than those Eidersoft scribes! Thank God I did not buy Archiver at the show, as it was not available. Instead of spending £15 on this useless package I should have spent it on four more cartridges.

Ralph A. Katthoefer,
Dimitrakopoulou No. 1, Voula,
Athens 166 73,
Greece.

/* I contacted Eidersoft about this and they say that you must have got a faulty cartridge, as no-one else has complained. They tell me that the package is now 100% machine code, and anyone who has problems may return their original version for replacement. Has anyone else got QLArt? */

USING MEMORY EXPANSION AND DISK TOGETHER

I have connected both my 512K memory board and CST disk interface in parallel (one into two connection) without any problems, apart from some tricky soldering. There appears to be plenty of power to supply both units. The difference in the operational speed of the QL using both these units puts it into another league. I have never found the Microdrives particularly slow, but because of the file buffering I find the disk unit to be the quickest I have seen.

For those with the same problem the DIN 41612 64-way (A+B rows) connectors are readily available from people like Watford Electronics. One word of warning - use plenty of insulation tape and treble check every connection (192 altogether) for

shorts and solder bridges. Ignore this at your peril!!! I used stiff wire as opposed to 64-way ribbon cable as the latter is not really Intended for intricate soldering, especially as the connectors are designed to be board-mounted. Having said that, though, my Mark 2 version will use Veroboard which will space the solder joints out, meaning that the ribbon cable should be better suited.

As I mentioned in my previous letter, it is doubtful whether this could be used to connect, say, the disk interface and a Centronics device. I haven't got the latter, so I can't say for sure, but in the meantime who needs Uncle Clive's Peripheral Expansion Unit - I don't!

John Lawlor,
35 Stonefield Drive,
Inverurie,
Aberdeenshire,
Scotland.

/* Miracle, who sell a similar unbuffered two-way adaptor, tell me that it only works properly with about 50% of QLs. */

REVIEW OF QC COMPILER

QC is supplied on two Microdrive cartridges, with a comprehensive manual in an A5 ring binder. One cartridge, labelled QC1, contains the compiler, an assembler, linker, linker control file, and backup programme. A second cartridge, QC2, contains the standard library, an additional QDOS library, a standard I/O header file, a programme called "compile" which drives the compiler, assembler and linker, a programme to alter the default windows used by the compiler, assembler and linker, the source code of the "compile" programme, and a backup programme. The cartridges are not protected. Two blank cartridges are also supplied, and a book called "A Book on C", by R.E. Berry and B.A.E. Meakings.

The documentation is pretty good and comprises the following sections:

- Introduction
- How to run the compiler
- The QC language
- QC standard I/O runtime library
- Extra QDOS library routines
- Interfacing with assembler code
- The command line and I/O redirection

Unfortunately, there is no index, but a summary of the library routines is provided, in alphabetical order, so it is not too difficult to find things. The trickier features (such as interfacing with QDOS, interfacing with assembler,

and passing the command line to a programme, are illustrated by examples. The manual is not intended to teach the C language, which is why the copy of "A Book on C" is provided. I don't think much of this book, and feel that users would probably learn more from the C programmer's "bible", Kernighan and Ritchie's "The C Programming Language". Unfortunately, I am told that this book now costs well over £20!

The compiler is a subset of the full Unix C compiler, and lacks features such as floating point, multi-dimensional arrays, structures, unions, typedef and bit fields. Most C programmers rarely use floating point, typedef and bit fields, although structures and multi-dimensional arrays are very useful. Multidimensional arrays can be simulated quite easily, however, by using arrays of pointers. QC appears to be upwards compatible with a full Unix C implementation (apart from the library functions specific to the QL), so it is ideal for learning the language. C is often used for systems programming - writing operating systems, text editors, assemblers and compilers - and QC is quite adequate for this sort of work. In fact, the QC compiler was written in QC, which should indicate its power. It is also ideal for writing simple utilities and filter programmes. The compiler generates 68000/8 assembly language, which then has to be assembled and linked to produce an executable machine code programme.

A very comprehensive set of options is provided, which are included on the command line. -M writes the first line of each function to the screen as it is compiled and -A beeps every time an error is found. A total of six options are available.

Like the original Small C, QC allows the use of inline assembly language code, using the `£asm` and `£endasm` directives. A limited macro facility is provided by the pre-processor, which is useful for defining constants and so on.

The compiler has one or two minor bugs, GST tell me, although I have only found one, which manifested itself when I gave a formal parameter the same name as its function, when the name had previously been defined as a global. This was rather silly of me, anyway! The compiler didn't catch the error, and the function just returned without executing.

No editor is supplied, as most users of this package will already have one of the editor/assembler packages. I use it with Metacomco's editor. The manual describes how to use Quill to edit QC programmes, which might be of interest to some users.

The standard I/O runtime and QDOS libraries are very comprehensive, and virtually all the QDOS facilities available from SuperBASIC are provided. A function to change the baud rate is not included however, which is a

strange omission. I wrote my own using the trap function provided - it only took seven lines of code.

The assembler is a "stripped down" version of the Sinclair assembler developed by GST. It accepts the output from the compiler and translates it into relocatable object code according to the SROFF (Sinclair Relocatable Object File Format) specification. Relocatable object modules produced by the assembler are then linked using the GST linker, to produce a programme that may be EXECed, in the usual manner. SROFF allows a library module to be added to, merely by concatenating files, and I have created my own library file, "mylib_rel", which contains several general purpose functions such as the aforementioned baud, a Shell sort, and several other useful bits and pieces. I have also written a programme called "concat", in C, of course, which makes it very easy to add additional library functions to my library file.

The "compile" programme supplied is menu-driven and makes the whole process of compilation, assembly, and linking, very easy. The programme uses overlapping windows for each operation, and looks very pretty. The source code is provided, which makes a nice programming example. A programme called "window_mgr" is also provided, which allows one to alter the default window configuration used by the compiler, assembler and linker, with the cursor keys, in a similar fashion to that adopted by Metacomco with their software.

Using the ""Sieve of Eratosthenes" prime generation benchmark programme described in Byte, August '83, I came up with a time of about 30 seconds for QC, which is a bit on the slow side. This is probably not the fault of the compiler, but due to the slowness of the on-board RAM in the QL, which effectively slows down memory accesses by a factor of about three. Running the programme in external memory should speed it up considerably, and I'll try this out when I get the new memory expansion module from Simplex.

Although Metacomco hope to have Lattice C available soon, which is a full implementation of the language, I feel that QC is probably better for the average user who wants to learn the language. It runs quite happily on the basic machine, although disk drives and additional memory speed things up considerably, whereas I think disks and expansion memory will be essential when developing software using Lattice C, although the programmes produced will run on a standard QL. QC is quite suitable for serious work, within its limitations, and one does not have to pay royalties for the use of the runtime library, if programmes are sold. This might not be the case with the Lattice product.

On the whole, I think this is a very good package, and reasonably priced at £59.95, which is about the same price as other low-cost C compilers for CP/M users, such as the Software Toolworks C/80, which does however have structures and multi-dimensional arrays.

Leon Heller

SIMPLEX MEMORY EXPANSION - AN UPDATE

Since last I wrote to you on the subject of the Simplex 256K memory expansion unit I acquired just before Christmas, my old one has been replaced by a new one which I have now used for Considerable continuous periods of time (although never more than 16 hours), and there have been no problems. In the circumstances.

I am entirely satisfied with it and the additional memory it provides.

W. D. R. Spens,
Marsh Mills Cottage,
Over Stowey,
Bridgewater,
Somerset TA5 1HG.

ABACUS

Some time ago (QUANTA, Jan. p 31) I described my efforts to solve the problem of stripping formulae from calculated values in order that they might be moved around the grid as absolute values. Whilst the method described works in my particular application, it is not error free in all situations, particularly if multiple copies are required, one of which needs further manipulation. I have now devised a much more elegant solution to the problem and one I believe to be completely error-free in all applications since it produces true numeric values.

1. Make a not of the grid co-ordinates of all the cells from which it is required to strip the formulae.
2. Load a new (blank) spreadsheet.
3. Put a zero value in each affected cell ensuring that the type (monetary, decimal, etc.) matches that of the cell to be modified.
4. Save this skeleton to Microdrive and Zap the display.
5. Load the spreadsheet to be modified.
6. Merge the skeleton just saved and the job is done.

Note. Don't try to save time by merging the spreadsheet into the skeleton, because it won't work that way round.

I'm still wondering why Psion couldn't have told me that:

Capt. E. D. Fieldson,
Miraflores,
Troutstream Way,
Loudwater,
Rickmansworth, Herts. WD3 4LA.

A FILE DUMP PROGRAMME IN BCPL

The following programme dumps a file in HEX and ASCII to the screen. It is very useful if you transfer a file to the QL from another machine, and you need to find out if it has any special characteristics, such as funny end of line characters, which might prevent you loading it into Quill, for example.

The programme asks for a file name, and then displays the file, 16 lines at a time, waiting for a key press before displaying the next 16 lines. Pressing <escape> aborts the programme.

/*

DUMP

BCPL programme to dump a file to the console in HEX and ASCII. Written by Leon Heller.

Version 1.1 - 30/4/85.

*/

Get ""libhdr"

Manifest \$(
LENGTH = 16
BYTES. PER. LINE = 8
ENTER = 10
ESCAPE = 27
MAX = 16
\$)

let start() be \$(

let fname = vec LENGTH
let line = vec BYTES. PER. LINE
let con.stream, file.stream, ch, n, i = ?, ?, ?, 0, ?
let lines = ?

screen(screen.border, white, l)
screen(screen.paper, black)
screen(screen.ink, white)
screen(screen.clear)
con.stream := findinput("con")
Writes("File name: ")
if readstring(fname) = 0 then stop(0)
file.stream findinput(fname)
if file.stream < 0 then \$(
Newline()
Writes("File open error")
stop(0)
\$)
selectinput(file.stream)

```
screen(screen.clear)
lines := 0
$( n read(line)
  for i = 0 to n do writef("%X2 ", line:i)
  screen(screen.tab, 26)
  for i = 0 to n do $(
    test line!! >= " " &line!! <= 'z'
      then wrch(line!!)
      else wrch('.')
  )
  if n < 7 then stop(0)
  Newline()
  Lines :=lines + 1
  if lines = MAX then $(
    selectinput(con.stream)
    If rdch()= ESCAPE then stop(0)
    selectinput(file.stream)
    screen(screen.clear)
    lines := 0
  )
  $) repeat
$)

and read(line)= valof $(
  let i, n= ?, 0
  for i = 0 to 7 do $(
    Line!! :=rdch()
    if line!! = endstreamch then resultis (n - 1)
    n :=n+1
  )
  resultis (n - 1)
$)

and readstring(v) = valof $(
  let n, ch = 1, ?
  $( ch := rdch()
    If ch = ENTER then $(
      v%0 :=n - 1
      resultis n - 1
    )
    v% n : =ch
    n :=n + 1
  ) repeat
$)
```

Leon Heller

OLIVETTI PRAXIS 35

Soon after ordering the QL an interface was ordered from Timtom Micro, costing £72.00. It arrived in February (84).

The Olivetti was opened up and the 25 soldered connections made as per Instructions.

The QL arrived in August..... No need for further comment!

I wrote to Timtom requesting confirmation that only three of the six wires in the cable lead required connecting. The same printed data was received together with a freehand sketch and a note saying "I am still waiting for my QL. I'd be grateful for confirmation of wiring to QL after you have fitted yours".

I didn't think it was funny.

The printer lead was severed and a 5 DIN plug rescued from an old tape recorder lead and the hard work started. TX Data, CTS Input and Ground would apparently be obvious from the QL manual. Being a novice it did not help much to know that the Baud rate required to be 300 when confronted by the number of variables in the printer driver. I forget when I discovered the printer driver. It took a week after discarding logic and perming any three from six in any order to make the correct connections. Never have I been so busily unemployed.

There can't be many who have equalled the achievements of Edison, Baird etc.

Elated I telephoned Timtom Micro. "He's not in today" was the reply. He wasn't in the rest of the week either and then I found a paying job and didn't try any more.

You may have gathered that the piece of equipment is not very elegant, but here is proof that it works and what's more it could be purchased before the QL.

Whoever the man is I would like to express my thanks and admiration.

P. W. Haragan
238 Plumstead Common Road,
London SE18 2SR.

If he still exists;
39 Bryn Gwyn,
Caerphilly,
South Wales CF8 1ES.
0222 885819

BEGIN AGAIN

I don't wish to discourage those who take the trouble to write for the benefit of beginners but they should take care to ensure that what they impart is correct.

Reading Frank Hope's item on Data Input, Quanta Vol 2 No 2, I was mystified on the merits of writing a procedure for the apparent purpose of shortening the keyword CURSOR by three characters. Surely code such as this can only obscure the proper use of SuperBASIC. The use of integer parameters in his DEFINED PROCEDURE INFO is contrary to the logic of SuperBASIC. The formal parameters assigned do not have a type since they automatically adopt the type of the parameter used when the Procedure is called; see the Manual - Concepts p 21.

The use of integer variables is claimed to speed processing but the Manual says there are no integer constants as all are stored as floating point numbers so where is the evidence for the claim?

Line 6005 is said to set-up the variables but it assigns a2% =w%. Where is w% assigned?

Line 6010 purports to initiate a REPEAT loop but as it is followed by a REMark it would operate as a short form and hence make an endless loop on this line as no EXIT statement is provided or could be used after a REMark.

I would question the wisdom of the use of END REPEAT repeatedly used after an IF condition.

Firstly it makes the construction of a repeat loop obscure and secondly one has to be very sure that only one of the IF's will be true; it was not at all clear to me that this would be the case.

The recommended method of exiting a repeat loop is to follow the condition by an EXIT statement, why teach something else?

The same poor construction of 'IF condition: END DEFINE' is used to exit from a DEFINED PROCEDURE instead of the recommended RETURN statement. Why?

I also question, particularly when instructing beginners, the use of non-expressive identifiers like r1 to r4.

I think it would also be more helpful if the text lines used upper case for the keywords (or parts of them) which will appear in upper case on the screen.

I consider that any sample constructions, such as those shown, should have been proven by the author to operate in the manner intended. Clearly this was not the case in this instance.

I appreciate that we all start from a different standpoint and there are many ways to kill a cat but I do suggest we adopt Sinclair's recommendations for SuperBASIC unless we can show good reason to the contrary.

John Tanner
43, Northumbria Drive,
BRISTOL BS9 4HL
0272-623401

RETRACTABUG

My item 'BUGS IN THE BASIC', QUANTA Vol 2 No 2, brought a response from Capt. Fieldson who had obviously read the Manual with greater care than myself. Concepts p21 on functions and procedures explains that a variable used as a parameter in a procedure call will be modified as the 'formal' parameter within the procedure unless it is enclosed in brackets in the call statement. My apologies to Uncle Clive. Any comments on the 'Print epidemic'?

On the other hand, my number formatting routine ,QUANTA Vol I No 12, brought no response so I took a further look myself with the resulting listing below. Whilst the methods are similar in several respects the principal difference lies in the avoidance of the division of the number to restore the decimal position being replaced by string handling in the hope of achieving better accuracy. Sadly I have been unable to demonstrate any improvement. It is now aesthetically better and it is easier to follow the logic. I am still open to comment if there is anyone out there!

Since writing my note about Rob Sherratt's contribution to starting the library I have had the opportunity to see the LIBFILE cartridge which is now available to members and documents the library beautifully. It was produced by Rob with, no doubt, a very considerable amount of effort for the benefit of IQLUG. I did have one problem with it however which I think helps to prove a theory of mine that there is natural law to the effect that there is always one more bug. The Libguide_doc states that the cartridge contains a database file called LIBFILE to be run under Archive. It didn't, it was called libguide_dbf, which accounted for the 'unable to open file' messages I obtained.

While talking of errors can I say, critically, that I hope that the software advertised by SNOWSOFT on the back cover of QUANTA Vol 2 No 2 is better debugged than the spelling in their advertisement. I counted five such errors!

And again on the subject of errors, my first draft of this item suffered the fate that Quill still metes out to the unwary. I tried to scroll back to the beginning before saving it and the system locked-up, requiring a reset. I acknowledge that Quill version 2.00 is streets ahead of version 1.03 but it bears little comparison, even now to an efficient WP! Does anyone understand why?

```
190   DEFine PROCedure NUMFORM2(n,p,l)
200     LOCal a,b,neg,t$: neg=n<>ABS(n): n=ABS(n)*10^p+.5: IF n<E9:
      n=INT(n)
210     n$=n: a='e' INSTR n$: IF a
220         b=n$(a+1 TO ): t$=n$(1 TO 1): IF a>3: t$=t$+n$(3 TO a-1)
230         n$=t$&FILL$( '0',b-a+3)
240     END IF : a=LEN(n$): IF a<p+1: n$=FILL$( '0',p+1-a)&n$
250     IF neg:n$='-'&n$
260     a=LEN(n$): IF p>0: n$=n$(1 TO a-p)&n$(a-p+1 TO)
270     a=1-a+p:IF a>0: n$=FILL$( ' ',a)&n$
280   END DEFine
290   :
300   DEFine PROCedure RESAVE
310 z$= 'mdv1_demnumform2_bas': DELETE z$: SAVE z$
```

John Tanner,
43, Northumbria Drive,
Bristol BS9 4HL
0272-623401

P.S Referring to P. K. A. Dyer's experience of Quill, every failing *he* mentioned was familiar to me. Psion took a very long time before admitting these shortcomings which made the matter worse. Does Leon have proof that most members found it satisfactory?

**'AN INTRODUCTION TO PROGRAMMING THE SINCLAIR QL' by R. A & J.W. Penfold
published by BERNARD BABANI at £1.95 ISBN 0-85934-125-9**

The authors state this book is for the beginner with no previous knowledge of computing and programming. An attempt has been made to complement the manufacturer's information not to just duplicate it

The result is 100 pages that provide an excellent lead into the QL User Guide. Starting with variables, arrays and string slicing, the book works through printing, windows, and control statements. Examples include sorting, sentence analysis and a guess the number game. A chapter on BEEP gives pitch values for 48 notes (alas different to those in QUANTA 2.1) and numerous sound effects programmes.

Structured programming is described with examples of print formatting for financial programmes and a metric converter. A chapter on graphics fails to cover the use of CURSOR/CSIZE/AT but is otherwise useful, with die simulation and screen doodling as examples.

An interfaces chapter describes the RS232C system in such detail as to reveal the authors' background in electronics. Finally there are chapters on number systems and channels/devices, the latter including a version of the PSION CLONE programme and a rather simple boot programme.

This book is excellent value. It contains more information than others three times the price. Recommended for beginners.

Gerard Phelan,
17, Gunnersbury Court,
Bollo Lane,
LONDON.
W3 8JL

**'Understanding the 68000' by Leon Heller.
Published by CENTURY COMMUNICATIONS at £7.95 ISBN 0-7126-0585-1**

A clear introduction states that this book is for "the microcomputer user who wishes to learn about one of the new generation of microprocessors" and for "the hardware enthusiast who delights in building his or her own system". The author states that no knowledge of assembly language is assumed.

The 108 pages start with the 68000 family architecture and an introduction to the concept of assembly language programming. The circuit diagrams of simple 68000 and 68008 systems to build are given. The full assembly listing of a machine code monitor for them is provided and its operation is described in some considerable detail.

The 68000 addressing modes are excellently explained together with some of the instruction set. However the example programmes, taken from the monitor, are not explained

line by line in the detail needed by machine code novices. An extra 20 pages of explanations and 'how to write from scratch' examples would much improve the book.

A long chapter details the 58 assembly language instructions, but does not give enough information for the newcomer to understand where a particular instruction should be used. The book has no index.

For QL users the hardware details will be of limited interest, They must judge the book on its ability to assist them to understand and write assembly language programmes for the QL.

The absence of any reference to QDOS calls and descriptions of the associated TRAP instructions means that another reference book would be required.

In summary this is an easy to read book that should be of value to its target audience. The QL user who is not also part of that audience may find a book dealing with assembly language and QDOS calls of greater value.

Gerard Phelan,
17, Gunnersbury Court,
Bollo Lane,
London,
W3 8JL

SOME NOTES ON ARCHIVE.

I have developed a fair sized database on Archive and have come across a problem not mentioned in QUANTA and not clear in the manual.

My database is ordered on one field and I regularly enter new records one or two at a time. However I ran into a problem where I could not enter any new records without getting an out of memory message despite being well short of the 714 possible records.

The entire key for every ordered field has to be held in the QLs memory and it was this space which was full. To my surprise I found that re-ordering the file on the same field completely cured the problem and I was able to continue entering new records until there was no more space on the Microdrive.

What I believe is happening is this. When a file is ordered the data is not swapped about on the Microdrive as this would be too slow. Instead a key is built which contains all the information of where the records lay. However when a new record is added it is not slotted into the key at the correct place, but tagged onto the end with some kind of flag in the main chain at the correct place which points to the new record at the end. After many new inserts the key has become very messy and occupies far more room than it should. Re-ordering tidies everything up.

If anyone has a more correct explanation of what is going on I'd be interested to hear.

As my database grew to more than could be held on one drive I had to split it across two. Below is the procedure that I used to do it.

P.S. The 'split_prg' file was copied as 'split_exp' and imported into Quill. It was printed on a Juki 6100 which by the way functioned first time with the CST interface.

```
proc SPLIT
REM VERSION 13.8.84 by Phillip Biggs
let DELLOOP=0
```


paper 0: ink 4: CLS : print at 3,18; "PROCEDURE TO SPLIT LARGE DATA FILES IN HALF."

input at 6,18; "Enter name of file to be split :-" ; FILE\$

print paper 6; at 5,0 " (put full line of spaces) "

print paper 6; ink 2; " TYPE 'YES' IF YOU HAVE TWO FULL BACKUPS OF CARTRIDGE NOW IN DRIVE TWO. ";

Print paper 6; " (All copies must have the same file name.) ";

Print paper 6; " (put full Line of spaces) ";

input at 9,36; TEST\$

if TEST\$= "YES"

print at 9,18; "Opening file"; FILE\$, " ".

open FILE\$

print at 9,18; "Number of records in file "; FILE\$, " = ";

let RECORDS=count(): print RECORDS

let FIRSTHALF=INT(RECORDS/2): let SECONDHALF = RECORDS - FIRSTHALF

print at 10,18; "Deleting first half of "; FILE\$, " file. Please wait."

while DELLOOP < FIRSTHALF

first : delete : let DELLOOP=DELLOOP+1

endwhile

close

while TEST\$<> "C"

print at 10,8; "Completed. Insert one backup copy in drive two and press 'C'.": let

TEST\$=getkey()

endwhile

print at 10,8; "Deleting second half of "; FILE\$: " file. Please wait. ";

open FILE\$

let DELLOOP=0

while DELLOOP < SECONDHALF

last : delete : let DELLOOP = DELLOOP+1

endwhile

close

print at 10,18; "Completed. Don't forget to make backups. "

else CLS

endif

endproc

It may not be very elegant but it works.

Phillip Biggs.

01 449 0036.

PRINTING FROM QL CHESS

After the usual delay, I have received my copy of QLChess, which seems to live up to its reputation. If other software writers will regard the three dimensional display of the chessboard as the touchstone by which their own work will be judged, we will get some very exciting things.

I have had a couple of problems, however, with the printing routine, and I offer my solutions, which may be of help to others. I use an MT-80 with Miracle Systems Limited's Centronics interface, and the worse problem was that I was unable to print the whole of the chessboard with <CTRL> P. I would get about half and then the routine would abort. Examination of the code showed that the problem seemed to be with the timeout used. The programme uses the QLDOS routine called by TRAP #3 with 1\$05 in register D0 to

send a byte to the printer. All TRAP #3 routines require a timeout byte to be placed in register D3 before being called, and the programme here uses \$1E, which produces a timeout of 0.6 second. What was happening was that about half the chessboard would fill the printer's buffer, and would then refuse to receive more bytes, and 0.6 second was too short a time for the position to be rectified, and so the programme would abort the routine.

I have solved the problem by replacing the timeout byte with \$FF, which is 5.1 seconds and should be long enough for all but the slowest daisywheel, but unfortunately a longer timeout is not possible without making major alterations. I used Tony Tebby's

invaluable files_code routines on the Library's first cartridge. and I suggest that after installing the routines, you should insert your write-enabled original copy of QLChess in mdv2_ and then write the following programme:

```
100 OPEN#3,mdv2_chessc
110 SET_POS#3,9727
120 BGET#3,a: PRINT a: STOP
130 SET_POS#3,9727
140 BPUT#3,255: CLOSE#3
```

Run this and it should stop after printing '30'. If it does not, check it and re-run. Otherwise, type CONTINUE <ENTER> RUN <ENTER>. This time it should stop having printed '255'. The other problem was that I was getting a blank line between every line of printing because, like most people, I should imagine, my printer is set up to receive <CR> and generates its own <LF>, while the programme was sending out both. The simple answer is to substitute <NUL> for <LF>, and this can be done with the programme above but with '7711' substituted for '9727' in lines 110 and 130, and '0' substituted for "255" in line 140. The first time the programme will print '10' and the second time '0'.

G. J. W. Cunliffe,
Oak Tree,
Chesworth Close,
Horsham,
West Sussex. RH13 5AL.

68000 MOVE Instruction Codes and QL Version JS

The letter from Bill Cowhig on CALLS from SuperBASIC illustrates a feature of the MOVE instructions which, in my opinion, is not always adequately explained or emphasised in some of the books on assembly language I have read. At the risk of boring the experts. I thought a short note on the subject might be useful to those who, like me, are struggling to master 68000 code!

The code in question was MOVE? #0,D0 where the question mark indicates the doubt whether it should be Q. N. or L. To explain the effect of each of the three instructions, I will assume that D0 initially contains the 32 bit hexadecimal number 89ABCDEF. After MOVE.L #0,D0, the contents of D0 becomes 00000000 as the assembler converts the #0 to a 32 bit zero long word. The actual code for this instruction is 203C 0000 0000 . After MOVE.W #0,D0, the contents of D0 are 89AB0000 as only the lower 16 bits are affected. and the code is 303C 0000.

The MOVEQ instruction, however, operates in a totally different way to the other two in that the immediate data is contained in the least significant eight bits of the Instruction word. The range of data is from +127 to -128(decimal), the negative numbers being in two's complementary form. The important feature of MOVEQ is that it is SIGN EXTENDED to a long operand and ALL 32 BITS are transferred to the data register. This means that if the most significant bit of the immediate data is 1, then all the other 24 bits in the register become 1. Thus after MOVEQ #-128,D0 (128 is decimal), the contents of D0 become FFFFFFF80 (80 hex=10000000 binary=128 dec). Taking our original example, MOVEQ #0,D0 will change D0 to 0000 0000 as the most significant data bit is zero. The code for this instruction is 7000, the immediate data being contained in the two lower order hexadecimal digits.

It can now be seen (I hope!) that for the code in Bill Cowhig's letter to work correctly, either the MOVED or MOVE.L instruction can be used, but not the MOVE.W as only part of the register is initialised.

The MOVEQ instruction can only be used for the transfer of immediate data to a data register within the range of +127 to -128(decimal), but is very fast as it only consists of one instruction word. Thus taking about a third of the time of the MOVE.L instruction. It should therefore be used in place of the MOVE.L wherever possible.

Changing the subject completely. I have Just received my QL back from repair for the second time and found that it has been updated to version JS with a 1.10 operating system. I gather from the weekly press that there is improved handling of Microdrives, amongst other things, and that it is not yet available in the shops (at 15/3/85).

The improvement is dramatic, as a 60K assembler file I have now only takes half the time to load that it did with the JM version. One of the Version 2 Psion programmes (I cannot remember which one), actually loaded on one occasion in an incredible 8 seconds (timed from the appearance of the screen title). With the great improvements being made in the supplied software and the QL itself, I think Sir Clive can now say, with a lot more justification than BR, That "We are getting there"!

A. M. Coppard,
128, Forest Drive,
Chelmsford, CM1 2TT
TEL: 0245 84270

Computer art!

Some time ago, about 12 years, to be precise, I came across my first sight of a small desk-top plotter. It was made by Hewlett-Packard and driven in parallel to a computer terminal at 300. baud. One could not help but to fall in love with the device as it was extremely easy to programme and connected to the computer in exactly the same manner as one of the ASR terminals with which it was also connected. Several programmes were written for this plotter and the most commonly used of these programmes was a small computer art programmer to generate patterns from revolving circles. The idea appealed to me and I developed the programme into a slightly more sophisticated version to investigate the patterns generated by revolving spheres instead of circles.

The theory behind the programme is that we define various parameters(attributes), about a sphere positioned in a 3-dimensional space. The sphere is defined to be located in a position x,y,z of value 0,0,0. Knowing this, we can define some properties of the sphere:-

1. r .. The radius of the sphere.
2. sr .. The starting rotational value.
3. se .. The starting elevational value.
4. ir .. The increase/step in rotation.
5. le .. The increase/step in elevation.

These values will define for us an initial and incremental point in 3-dimensional space. We repeat this process, using the defined point, for a second sphere, centred at the point defined by the previous sphere. This will give us the end point of a second sphere. This process is repeated for a third and fourth sphere. Bear in mind that any of the spheres could take a zero radius and would therefore have no effect on the result. -The final data to be supplied are the axes from which the pattern will be viewed. In our model we label the axes X,Y and Z and choose any two of these to be the face from which we view space. It is quite possible to plot a pattern from one particular view in space and then input the same data and look at it from another face. This often gives quite startling results.

Having defined all this information, we find that we can, by using basic mathematical theory, calculate the position of the moving end-point of all these spheres. If one chooses the points carefully, the resulting pattern can be extremely beautiful. I have supplied Leon with some copies of patterns produced on my own Epson HI-80 plotter. A copy of the programme for both Epson HI-80 plotter and a screen version are available from the software library without charge.

TIPS.

When using the programme, make the increase in elevation of the first sphere a very small number, as this is the number of units used to step through the pattern. Ideal values are -1.0 to +1.0

Separate all input data by the 'ENTER' key.

In order to give you a start, here are some examples of pattern which look quite good on the screen:-

```
1000      0      0      178      1
1000      0      0      -1      1
0         0         0         0
 X       Y
1000      0      0      179      1
1000      0      0      1      1
0         0         0         0
 X       Y
```

V. R. Dwight,
T:mi Softronic,
Suvikuja 3B14,
02120 Espoo 12,
Finland.

Example Programmes for the 68K/ASM Assembler

Chris Scheybeler of GST Computer Systems Limited tells me that the latest versions of the assembler being distributed contain example programmes. I have received from him two example programmes: (a) SPACE.ASM - source of the utility programme that determines the amount of free space on a named directory, and (b) ECHO.ASM - a programme which reads and displays text from a specified file to a defined window. I will be glad to make copies available to any IQLUG members who purchased their 68K/OS assemblers before a source programme was included, and who care to send me a blank tape with SAE.

While the programmes do not add anything to the facilities of the system they illustrate a number of 68K/OS system calls and are examples of programming style worth copying.

I am still hoping to be able to set up a IQKOS library, but we need to have useful routines to put in it! So far response has been low. If you have anything to contribute please send it to me on tape (which will be returned as soon as I have copied it).

Norman Macleod,
Mount Pleasant,
Lea Bailey
Ross-on-Wye HR9 5TY
Tel: 098 981 494

SERIAL NUMBERS etc.

I recently had occasion to return my machine for the second time to have a self-motivating Microdrive cured, and I was pleasantly surprised to have it returned repaired within ten days. I also had my postage refunded, but not so far the update of the User Guide which I requested at the same time. I am intrigued to read in some communications to QUANTA that machines are returned with the serial numbers changed. Surely these are different machines from a later production batch and probably another batch of faults in them. That is why I was particularly glad to get the same machine back again, apart from the errant Microdrive it has shown no vices. I have not noticed anyone report that the initials which come up after the version number when Psion packages are loaded, are machine dependent and not Psion version dependent, although I don't know what their significance is.

USE OF PSION PROGRAMMES

Like other members I have learnt the hard way that it pays to use a system when employing the Psion programmes. I have developed one which minimises the anguish when things go wrong. I keep one cartridge solely for use in mdv2 when a Psion programme is in use in mdv1. All copying to and from this cartridge I do by direct commands in BASIC. This will sound tedious, but its a lot less tedious than repeating all the work when a whole cartridge is wiped as has happened to me in both Quill and Archive (though the second occasion was due to the previously mentioned Microdrive and probably no fault of Archive). I found the note on the capacity of Archive files very useful, it arrived at a time when I was compiling a large index of nearly 3000 records. This eventually reached eleven files, but I might be able to combine some of these. One puzzle which I have not been able to solve yet is how to order a field into a single alphabetical list if I have a mixture of upper and lower case as first letters

“BASIC PROGRAMMING ON THE QL”

Whilst I was QLless I bought a copy of the Cryers' book and agree with Leon, it is very good book. It probably has some tips even for experienced BASIC users, although it is not an advanced book. The one criticism you noted, Leon, is in fact down to you, the information you couldn't find is at the top of page 62.

/* It isn't, I'm afraid! At the top of page 62 it just mentions string variable names terminating in a '\$', nothing about names of functions that return a string terminating in a '\$', which was my grouse. */

MONITOR CONNECTIONS

My machine arrived back at about the same time as QUANTA 2_1 and noticing A. W. den Hartog's letter I thought I would try his connections on my Zenith ZVM-122 to see if there was any improvement in the display. In fact it turned out to be unusable.: the picture was worse but more important I no longer had a cursor, and could not communicate with the screen. If A. W. den Hartog really does want a black case for his monitor, the simplest and cheapest way of getting one would be to paint it.

SMITH-CORONA D200 PRINTER

First class, read the manual, set the dip switches, and then print. The manual is one of the clearest I've met, in spite of having a rather unusual arrangement of two A5 pages side by side on an A4 sheet. I've produced a sampler of 59 typefaces and that's not exhaustive. This letter is printed in pica standard NLQ, and I think you will agree the NLQ is very good indeed. It would almost be justifiable to drop the N. The printer will also support proportional spacing, but it would be unfair to expect Quill to cope with that and right justification. The printer is so versatile that there seems little point in sending a specimen 'install_bas' however if anyone would like any help they can't get out of the manual they are welcome to contact me on 041-942 5268. I should mention that I purchased the printer from Microworld in Brighton by mail order after first ascertaining by telephone that they had one in stock. There is also a D100 which is much the same but does not support NLQ.

PSION CHESS

Has anyone any experience of this? By all accounts it is first class version, but the only extended review I've read was in 'QL User'. In that it was reported that the programme could be copied and the copy used in mdv1, but the master had to be resident in mdv2, otherwise it wouldn't run. Now that seems to be asking to have your master corrupted to have it in a drive whilst the machine is under programme control. I would want to be able to make a security copy and then put the master away safely.

Ronald Besant,
24, North Grange Road,
Bearsden,
Glasgow G61 3AF.

/• In answer to your query about the Psion chess programme, Psion tell me that provided at least one sector on the master cartridge is uncorrupted, it will work OK in drive 2, so it is very unlikely for the master to become unusable. */

Aylesbury QL_User Group

At present, there doesn't seem to be a local user group specifically for the QL. in the Aylesbury area. I would like to set up a local group. If there are IQLUG members in and around Aylesbury who, like me, would like to get together with other QL users once or twice a month, please contact me. If enough are interested, then a local school is willing to - 'host meetings of QL users'.

David Nowotnik,
12 Long Plough,
Aston Clinton,
Aylesbury,
Bucks,
Tel:- Aylesbury 630067

USING QLCHES WITH ADDITIONAL RAM

A couple of points which should be of interest to other QL users. Firstly, with a 512K RAM pack attached PSION Chess won't run using the standard BOOT programme provided! This is because QDOS In its infinite wisdom gulps up 12.8K for Resource Management / System Variables with this amount of memory available resulting in Error messages when using RESPR at a low address. The obvious answer is to locate Chess higher up in memory but this creates more problems when Chess is running - the m/c isn't completely position independent for some strange reason. Whilst talking about PSION's quirks why when they've gone to the trouble of writing the version 2.00 software around QDOS haven't they gone the whole way and used full (flashing) cursor enable to at least allow Quill if not the others to fully multitask with other m/c software (screen refresh is available using F2) for people with more than 128K memory? Quilt also allocates itself QDOS memory in some very peculiar ways - at this very moment it has allocated itself approx. 325K of Transient Programme area and 265K of Heap Space !?! I'm sure they can do better than this but perhaps they're afraid of the OPD / XCHANGE market dwindling.... Anyway, to return to the original problem the only answer is to remove the RESPR statements In the BOOT programme and load the m/c 'chessc' straight into the area starting at variable 'addr' - not recommended in any textbook but just by good luck it works, without crashing the machine. With regard to cloning the cartridge I've managed to transfer Chess to floppy' Disk removing the unmentionable check to mdv2_ but had to use machine code. Has anyone done it successfully from BASIC?

The second point concerns the QDISK Interface and it's lack of compatibility, in my case with Metacomco's BCPL. I first suspected a bug In the QDISK when a file header reader I wrote which used FS.HEADR (TRAP#3, D0=47) returned an error every time when attempting to read a file from Disk. Removing the error check enabled the read to be carried out apparently without error! Anyhow the BCPL package appears to make FS.HEADR error checks (whereas the Metacomco Assembler package doesn't?) and wouldn't run from Disk. All parties concerned are aware of the problem and CST will exchange your old 1.04 version interface for a 1.06 version at no cost - problem solved. Note that as supplied QDISK is NOT compatible with BBC Disk drives until you remove the polarising lug (removed for me by Computamate using a file) from the drive cable plug so that It can be inserted into QDISK upside-down someone must have used a mirror when designing the PCB.

Nice one, OEL. They debited the money for the QCOM package from my Access account at the beginning of February, forwarded no QCOM and called in the receiver In mid-March! words such as Fraudulent Conversion spring to mind... I understand from a brief conversation with Access that I am covered by them against such an occurrence and I've put the matter into their hands but there must be a lot of people reading this who sent cheques.

John Lawlor,
35, Stonefield Drive,
Iverurie,
Aberdeenshire,
Scotland.

MERGE OVERLAY TECHNIQUE

At our recent local meeting Ken Otway, who had been showing us his recently acquired disk interface and drive, threw out the remark that I should ensure that the programmes I was writing took account of alternative storage media. I had written a boot programme to load the 'exten_code' which I use and to give easy selection of the required programme (Automenu on Library) but I had particularised it to use Microdrives only. I had also written a simple Customize programme for users to introduce their own headings for hardcopy output and to

Install their printer so I began to think where to start making the fixes. It first appeared that there was something of a 'catch 22' since one could write a file with the default device name in it but unless the system knew where to look it could not read the file but then I realised that QDOS has the answer in that it boots to the default device and loads any 'boot' programme so that provided just the 'boot' programme is modified by the user to contain the device name it can then be held in memory under a variable name. So far so good but if we do an LRUN from the boot programme we clear all variables so it had to be an MRUN to retain memory. When an MRUN is made the programme called starts from the next numbered line after the RUN statement. In order to avoid carrying forward a lot of unnecessary code one must either ensure that the new lines will blanket the boot programme (by having matching line numbers) or a DLINE of the calling programme should be done before the call. At first I could see no alternative to resurrecting the defunct GOTO statement to return to the DLINE statement. Then I realised that the RUN statement is no more than a particular GOTO and is not entirely beyond the pale so I saw the solution which is clearly illustrated by the three short demonstration programmes listed. It can be seen that, if for purposes of memory conservation, it is desired to split the working programme into a series of overlays the first programme segment can chain the next segment and by suitable assignment of the 'chain\$' variable one can proceed to the next or go back to any previous segment. If there are common procedures to be used which are common to more than one segment, storage space and loading time can be optimised by DLINEing only that part of the first segment which is not required for use in the second and subsequent segments.

Using this method I am developing a programme, which requires the largest possible arrays to optimise its area of application, in four separate segments, the first containing the basic assigned variables, the fixed arrays and the data for same which hence take up no programme space when later segments are called. A further incidental advantage is that as the screen is not cleared during a MERGE or MRUN, unlike a LOAD or LRUN, a suitable display can be maintained to give the user information on just why he is waiting. My only fear is that a 'not implemented' error could arise when the DLINE statement is met. It has happened once and QDOS behaves oddly when memory is short!

This may sound a clumsy solution to you ROM peekers so if someone can tell me where I can identify the default device without asking the user to make his own fix I should be delighted.

Changing the subject, and at the risk of boring you, I came across an interesting variant on the number formatting routine. It was contributed by a Bill Harper of Tiverton to the ICPUG newsletter and hence written in Commodore BASIC. It did not consider negative numbers and only allowed for tabbing on screen rather than creating a fixed string length, which is necessary when hard copy may also be required, so using his logic I re-wrote it in SuperBASIC including allowance for negative numbers and padding with spaces which extended it from his single line to two. It has a restricted area of application as it does not cope with large exponent forms, i.e. $\pm 1E6$ and greater, and is limited to a precision of between three and five places according to the size of the number. Since this almost restricts its use to money precision, for which it was intended, I have also written a particularised form for positive numbers rounded to two places of decimals. This is still restricted to numbers less than $1E6$ but the routine previously published could be stitched in at the front to normalise the number first if required.

I promise not to mention number formatting again.

```
1 BOOT:DLINE 100 TO 9990: MRUN chain$
2 RUN
100 DEFine PROCedure BOOT
110 REMark boot programme
120 dev$= 'mdv1_':PRINT 'boot prog':chain$ = dev$& 'chain1'
130 END DEFine
140 DEFine PROCedure RESAVE: z$ = 'mdv1_boot': DELETE z$: SAVE z$

100 DEFine PROCedure BOOT
110 REMark chain1 programme
120 PRINT 'chain1 programme': chain$=Dev$&'chain2'
130 END DEFine
140 DEFine PROCedure RESAVE: z$ = 'mdv1_chain1':DELETE z$: SAVE z$
```



```
100 DEFine PROCedure BOOT
110 REMark chain2 programme
120 PRINT 'chain2_prog': chain$= dev$& 'boot'
130 END DEFine
140 DEFine PROCedure RESAVE: z$= 'mdv1_chain2': DELETE z$: SAVE z$
```

```
100 DEFine PROCedure NUMFORM3 (n,p,l)
210 LOCAL a,t$: REMark ( NUMBER,pRECISION,lENGTH )
220 a=ABS(n)+.5/10^p: n$=INT(a): t$=a-INT(a)+l: PRINT t$: IF n<0: n$= '-'& n$
230 t$=t$(2 TO p+2): a=LEN(n$): n$=n$&t$: IF a<l: n$=FILL$(' ',l-a)&n$
240 END DEFine NUMFORM
250 :
260 DEFine PROCedure NUMFORM4(n,l)
270 LOCAL t$: n=n+5E-3: n$=INT(n): t$=n-INT(n)+1: n$=n$&t$(2 TO 4)
280 IF LEN(n$)<1: n$=FILL$(' ',1-LEN(n$))&n$
290 END DEFine NUMFORM
```

John Tanner,
43, Northumbria Drive,
Bristol. BS9 4HL
TEL: 0272-623401

ARCHIVE BUG

There seems to be a bug in ARCHIVE Version 2.0 which could be catastrophic for a business. Please bring it to the attention of IQLUG members.

When using ALTER, the record being altered occasionally duplicates itself and the first record and the file gets deleted in the process. By the time I discovered this, I had lost several records and copied the defective files on to my backup cartridges. It was pretty difficult to reconstruct the missing records.

Since the fault is intermittent I have not been able to find out exactly when it occurs, but it may have something to do with altering fields that are longer than the screen width. It's possible that this is the only situation in which it arises, but I can't be sure. Apparently what happens is that the record being altered overwrites the first record on the file, which must presumably be in RAM, and this is then re-ordered into sequence.

As a QLUB member, I wrote to Psion about it. Their reply did not mention the problem at all, going on instead about LOCATE and NEXT not working in an ordered file, a problem which I haven't had. They also sent a patch programme for the LOCATE and NEXT problems. I've tried it, but it makes no difference to the ALTER problem, so I've written to Psion again. A copy of my description to them is enclosed.

FOLLOW-ON ENQUIRY REF BS/0001779

PROBLEM:

In ARCHIVE version 2.00, when using ALTER, the record being altered duplicates itself and deletes the first record on the file. Experiments suggest that when this happens, the record being altered overwrites the first record of the file (which must presumably be in RAM and this is then replaced in file order, next to the altered record. (The experiment involved creating some records at the beginning of the file which were identical from the ordering point of view, and altering a non-key field in one of them. The duplicate record then turned out to be the first in the file, rather than being next to the one which had been altered.)

CIRCUMSTANCES:

File of 140 records, with 14 string fields, ordered on two fields. If I perform the following operations repeatedly, the problem occurs sooner or later:

PSION UPGRADES

I would like another opportunity to comment on whether or not the software upgrades should be free, in continuance of my letter which you printed in the March issue of Quanta. Firstly I would like to reply to your suggestion that my vers 1.03 QUILL tape might be corrupted by pointing out that I had five replacements and noticed no differences; and I certainly would not go into print on the basis of experiences with a single tape. Secondly I would compare the member who successfully wrote a 10,000 word thesis with a million pound winner of the Pools; in my experience I would suggest there is about the same probability in both cases.

I have rooted out from my scrap paper some examples of the things which went awry, and I enclose them; the first example is part of a document which has been printed out with about the first twenty lines on page 1, six lines on the bottom of page 2 which continue on to the top of page 3 for a further ten lines and then the rest of the page is blank. The second example is similar with four lines at the top of one page but with just two words towards the bottom of the next page, this is followed by a blank page, four lines on the next page and then a further blank page. The third example shows a different type of corruption where the structure of the document has been lost, that is to say that paragraphs no longer exist, the sentences run one after the other and the justification is no longer operating which means that words are split at the right hand margin. The fourth example shows that Quill vers 2.0 can also corrupt by inserting spurious line feeds.

When I sent similar examples to Psion (Relating to vers 1.03 only) I received the following reply:

Problem Reference : 0001028

QUILL sends unwanted line feeds or causes odd results when printing. If the DESIGN option DISPLAY is set to 4 or 6 and the PRINT command is selected QUILL can cause unwanted line feeds or other unwanted control codes to be sent to the printer. To avoid this problem always set DISPLAY to 8 (for 80 column) prior to printing. Altering the margin will also sometimes cause this problem.

NOTE: QUILL WILL SET DISPLAY TO 4 (40 COLUMN MODE) IF F2 IS PRESSED ON BOOT UP

This fault alone should warrant a free replacement.

It is interesting that even QLUB magazine has included statements to the effect that QUILL is not usable with documents longer than a thousand words, and that at least 40 free sectors are required per thousand words for successful use.

In addition to my comments on QUILL I would like to include a few general facts, extracted from Psion's 'Problem' replies to me: The failure of files to read could be due to the Microdrives overheating; cartridges may not be 'portable', that is they may not work in all drives; if you cannot clone a programme cartridge, coming up against 'bad or changed medium', obtain DIR_MDV1 and delete the last file then COPY that file individually and if successful COPY the remainder Individually. (Who has the time?)

One of your correspondents suggests that, in his opinion, the QL is a low priced machine and therefore one should accept the faults and not expect too much of it, and draws a comparison with the cheaper end of the motorcar market. I happen to be in the cheaper end of the car market and I know exactly what I expect; a vehicle which works satisfactorily and meets its specification but does not provide the luxury and refinement of the more expensive models. The QL, in my opinion, does not work satisfactorily nor does it meet its specification, and if it were not for the fact that I am

Heavily committed with files in QUILL and ABACUS I would be after a refund in order to buy a different machine.

By the way this is the third attempt to write this letter, using QUILL vers 2.0, as the two previous efforts have suffered from a locked-out keyboard! In addition I lost the complete screen whilst cleaning up my errors and after receiving a 'short of memory' message which claimed to be saving to Microdrive, but which did not even start Mdv2_.

Peter Dyer,
Kineff House,
Devon Bonk,
Guilford,
Surrey. GU2 5DQ

Machine Code Extension to Read Current Graphics Coordinates and Turtle Heading

The following code adds a function WHERE ([#n],n) to SuperBASIC. The channel number defaults to #1 if it is not entered as the first parameter. The other parameter must be 1, 2 or 3. If it is 1, the Y coordinate is returned, if it is 2 the X coordinate is returned, and if it is 3 the Turtle Heading is returned. The numbers returned are floating point format. The code for identifying the channel table is taken from A. Dickens (QL Advanced User Guide), page 294. The first 4 bytes of each table are the channel ID. Next come 3 floating point numbers (6 bytes each), the Y coordinate, the X coordinate and the Heading respectively. I shall be sending this, and another version which also has a function to determine whether the Turtle PEN is up or down, together with some Turtle graphics programmes which make use of them, to the IQLUG Library.

*Operating system vectors

```
CA_GTINT EQU $112 ;Gets arguments as integers
BP_INIT EQU $110 ;Routine to link new function to BASIC
ERR_BP EQU -15 ;'bad parameter' error
ERR_NO EQU -6 ;'channel not open' error
```

*Operating system offsets and equates

```
BV_CHBAS EQU $30 ;offset from A6 of pointer to channel block
BV_CHP EQU $34 ;offset from A6 of pointer to end of block
CH_LENCH EQU $28 ;length of one channel entry
BV_RIP EQU $58 ;offset from A6 of RI stack pointer
RET_FP EQU 2 ;Function return Floating point number
```

*

*First initialise the BASIC function WHERE ([#N],n)

```
MOVE.W BP_INIT,A2 ;link procedures and functions
LEA PROC_DEF,AI
JSR (A2)
MOVEQ #0,D0
RTS
```

*

*BASIC procedure definition: No procedures, 1 function - WHERE

*

```
PROC_DEF
DC.W 0 ;Number of procedures
DC.W 0
DC.W 1 ;Number of functions
DC.W WHERE-* ;Start of code
DC.B 5 ;Length of name
DC.B 'WHERE' ;Function name
ALIGN
DC.W 0
```

Function to find the current Graphics Co-ordinates

- * Set default or given channel in D6
- * A3 and A5 are standard pointers to name table for parameters.
- * A1 is used as a pointer to the arithmetic stack.
- * The value is returned to BASIC via the arithmetic stack.
- *

WHERE

```

MOVE.L    BV_RIP(A6),-(A7)    ;Save arithmetic stack pointer
MOVEQ    #1,D6                ;Default is channel #1
*
BTST     #7,1(A6,A3.L)       ;Did the first parameter have a # sign
BEQ.S    SET_CHANNEL         ; ... no if zero
.
*
if there is a # then the channel number must be changed.
*
MOVE.L    A5,-(A7)           ;Save pointer to top of parameter list
MOVEA.L   A3,A5              ;Set new top to be
ADDQ     $8,A5               ;8 bytes above A3
MOVE.L    A5,-(A7)           ; and save for next parameter fetch
MOVE.W    CA_GTINT,A2        ;Routine which puts integer on stack
JSR      (A2)                ; and execute
MOVEA.L   (A7)+,A3           ;Reset parameter pointers
MOVEA.L   (A7)+,A5           ;These MOVEA Instructions do not reset
                                           ;condition codes
BNE.S     EXIT_BP            ;if call was not successful then error exit
MOVE.W    0(A6,A1.L),D6      ;move return from stack to D6 to be new
                                           ;channel number

```

*Now point to correct BASIC channel table

```

SET_CHANNEL
MULU     *CH_LENCH,D6        ;Multiply channel no. by length of entry
ADD.L    BV_CHBAS(A6),D6     ;Then add the standard offset to channel
                                           ; table relative to A6
CMP.L    BV_CHP(A6),D6       ;Check that it is within table area
BGE.S    EXIT_NO             ;No, then channel not open
*
* Now get second parameter which decides whether the x, y co-ord or heading is wanted.
* Y is 1 : X is 2 : heading is 3
*

```

```

MOVE.W    CA_GTINT,A2        ;Set vector
JSR      (A2)                ;Get Integer
CMP.W     #1,D3              ;Check one parameter returned
BNE.S     EXIT_BP            ;No .. then error exit
MOVE.W    0(A6,A1.L),D5       ;Move return from stack to D5
CMP.B     #1,D5              ;check parameter is between 1 and 3
BLT.S     EXIT_BP
CMP.B     #3,D5
BGT.S     EXIT_BP

```

* Now point to required co-ordinate or heading - all floating point numbers

```

MULU.W    #6,D5              ;times 6 for 6 byte offset
SUBQ.W    #2,D5              ;subtract 2 as channel ID is only 4 bytes
ADD.W     D5,D6              ;and add to channel pointer
MOVE.L    D6,A3              ;Now put in address register for next op.
MOVE.L    (A7)-,A1           ;Recover stack pointer
SUBQ.L    #6,A1              ;Set space for FP number on stack
MOVE.L    0(A6,A3.L),0(A6,A1.L) ;move 1st 4 bytes of FP to stack
MOVE.W    4(A6,A3.L),4(A6,A1.L) ;and last 2 bytes
MOVE.L    A1,BV_RIP(A6)      ;Reset arithmetic stack pointer
MOVEQ     #RET_FP,D4         ;Set D4 to indicate type of return
MOVEQ     #0,D0              ;set D0 to 0 to indicate no error
RTS

```

```
* error exits
EXIT_BP
  MOVE.L    (A7)+,D0          ;Saved value of RV_RIP is not needed on
                             ;error return but user stack must be tidied
  MOVEQ    #ERR_BP,D0       ;Gives 'bad parameter' error on return
  RTS
EXIT_NO
  MOVE.L    (A7)+,D0          ;As above
  MOVEQ    #ERR_NO,D0       ;Gives 'channel not open' error on return
  RTS
*
END
```

Printing Arrays

I have recently discovered that array slicing works within a PRINT statement. If the array name alone is used in the PRINT statement then all the elements of the array are printed. The normal print separators (; , ! \) all seem to work as well, so formatting can still be accomplished so long as it follows the same pattern throughout. Slicing also works, so PRINT array (TO 20) will print all the elements of the array with a first subscript up to 20. Since you do not have to set up your own loop (e.g. a FOR loop) it is faster to print arrays like this. Certainly it is useful when you are debugging a programme.

David Coles,
3, Boothey Close,
Bigglesvade,
Beds, SG18 0DG.

**'SINCLAIR QL User Guide' by Lionel Fleetwood
published by SIGMA PRESS at £7.50 ISBN 0-905104-92-7**

There is an impressive and detailed structure to this 208 page book. After an introduction to the QL and SUPERBASIC there is a set of chapters on each of the PSION packages, followed by appendices. The typeface is small and a lot of information is conveyed. There is an index containing SUPERBASIC keywords but little else.

The SUPERBASIC section is a bit of a washout. It does not try to describe much of the language, referring you instead to another book. It does tell you about default Microdrive references in LOAD/SAVE, INVERSE, INPUT....TAB, PRINT....AT, DO....WHILE, LGO and FRE(), none of which exist in current machines.

The PSION sections take a quite different approach to the user manual. The author has clearly tried hard to work out some helpful examples. In this he has succeeded. However there are errors in these sections too. Hence we are told EASEL will print only onto a colour printer.

Readers whose English vocabulary is limited to BASIC keywords may balk at BRICKS, CURCONS, WRAITHS and CONGENERs. However in using these words the author has clarified many difficult concepts and his writing style has given the book personality and warmth.

A second, corrected, edition may be of value to any QL users, who find the supplied PSION documentation unfollowable. It is only a little simpler though, so those quite lost may be better off with other books. This edition has too many errors to recommend purchase.

Gerard Phelan,
17, Gunnersbury Court,
Ballo Lane,
LONDON.
W3 8JL

Notes from your secretary.

Please note that you can ring up to 10: 30 pm. with lunches 12:45 to 2:15 and teas 5:15 to 6:15. Sometimes a chat may be out of the question' due to current workloads. I am sorry for delays on supplying local members but I tend to do these requests in a batch.

Please keep the label on your envelope as this is to be used for library purposes and claiming discounts from suppliers. Please note that disks are now available for under 10 pounds for 10. Ring Leon for contact's number.

Chas. Dillon's 100 pages on ARCHIVE have received a rapturous reception and Rob. Sherratt has prepared 10 pages of notes on how to use the LIBFILE cartridge which contains details on the library..
Chas. Dillon's Notes £3 75
Chas. Dillon's cartridge £5.00 or if cart supplied £3.00.
Library 'hard COPY' 50 pages £2.75.
Please send in stamps with requests for the above.
Sid Day can supply formatted tapes with the library programmes already saved.

SUB-GROUPS and contacts:

Aylesbury	David Nowotnik.	
Birmingham	Andrew Fitton	021-477-4618.
London North	Ashok Soni	01-472-9966.
London South	Don Forbes	01-688-5794.
London West	Gordon Henson	0-373-4647.
Bristol	John Tanner	0272-623401.
Oxford	Francis Cameron	0865-240058.
Southampton	Jonathan Kempston	0703-766879
Liverpool	Peter Stewart	0704-37626
Edinburgh	Ted Gladstone	031-661-4558
Sweden	Kalle Blomqvist.	
	Mbrodsaatan	17,5-431 42 Molndal.
Abacus	John MacDonald	Manchester.
Archive	Chas Dillon	
	(Prof superb!)	0827-897343.
Archive	Sandra Essex	08831-4796.
Cash Trader	Albert Morris	Stockport
QUILL		required.
Abacus		'assist' required.
Easel		required.
General (eg BASIC. ARCHIVE Systems		
	Gordon Henson	see above.
Machine Code	Gary Williams	01-907-2360
Head Librarian	Sid Day	0708-27272

As these notes will appear here every month will area contacts 'Pester' me to ensure inclusion.

WORKSHOPS

JUNE 2nd, London Kennedy Hotel by EUSTON STATION
This will be a repeat of the disk interface workshop held in January. Space will be at a premium and bookings are essential. Trade tables cannot be entertained on an ad hoc basis as we have the use of one room only. The cost will be £5.00 and long distance travellers will be entitled to discounts. Ring for details.

June 30 London Kennedy Hotel by EUSTON STATION
This will a 68000 hacker's day and it is expected to be hardware dedicated. Don Forbes suggested the idea and the cost will be £3. 00. Will those who are bringing something please ring me and there is no reason why for a for sale/wanted cannot be run. Trade tables will be available at £10.00 and must be pre-booked. There might be one or two jam sessions on the lines of the GERT OELKERS and others 'set' and enjoyed at Swindon in March.

ARCHIVE in view of the nice comments Chas Dillon has generated, will members please ring me if they would like a day on ARCHIVE with Chas. Venue will be Kennedy Hotel, Milton Keynes or Birmingham.

ABACUS ...{original unreadable}... 'be in August' and almost certainly in LONDON and if so then at the Kennedy Hotel.

Brian Pain.

NEW HORIZONS
SOFTWARE
Present

SOFTWARE FOR THE SINCLAIR QL-

CHOMPER £10

A VERSION OF PACMAN WITH GHOSTS, POWER PILLS, ETC.

ZFRED £10

A VERSION OF QBERT, PAINT THE PYRAMID BEFORE THE SNAKE AND BALLS KILL YOU.

GOLF £10

A FULL SIMULATION OF AN 18-HOLE COURSE, BUNKERS, WATER, CLUB SELECTION, ETC.

STRANDED £10

YOU ARE STRANDED ON A HOSTILE PLANET. CAN YOU REACH THE TRANSPORTER BEFORE YOU ARE KILLED BY MUTANTS? (AN ADVENTURE)

SHOOT 'EM UP £10

AN INVADERS VARIANT. 20+ LEVELS

FROGGER £10

A VERSION OF THE ARCADE GAME.

CRYSTAL COLLECTOR £10

CAN YOU SURVIVE IN YOUR SEARCH FOR THE PRECIOUS CYTON 77

DUNGEON £10

AN ADVENTURE - CAN YOU ESCAPE FROM THE DUNGEON AND SAVE YOUR FELLOW VILLAGERS FROM THE ORCS?

THE GOLDEN CHALICE £10

YOU MUST VENTURE INTO MANY STRANGE LANDS IN SEARCH OF THE GOLDEN CHALICE. (AN ADVENTURE)

DAMIEN £10

CAN YOU DESTROY THE SPAWN OF SATAN BEFORE HE TAKES OVER THE WORLD? (AN ADVENTURE)

ELFIN WAY £10

YOUR TASK IS SIMPLY TO SURVIVE AS AN ELF IN THEIR SHADY LAND.

SNOOKER £10

A FULL SIMULATION OF THE POPULAR GAME.

MARTIAN AFFAIR £10

YOUR TASK IS TO FIND AND DEFEAT THE DRAGON. (AN ADVENTURE)

COUNTDOWN £6

A VERSION OF THE POPULAR T.V. SERIES.

NEW HORIZONS
SOFTWARE
Present

SOFTWARE FOR THE SINCLAIR QL-

BEACON ADVENTURE £10

YOU MUST FIND AND LIGHT THE BEACON IN ORDER TO WARN YOUR FELLOW VILLAGERS OF IMPENDING DANGER.

HOME ACCOUNTS £15

AN ACCOUNTS SYSTEM TO HELP MANAGE THE HOUSEHOLD BUDGET.

TIME SLIDER £10

CAN YOU FIND YOUR WAY BACK TO YOUR OWN TIME? (AN ADVENTURE)

PROGRAMS1 £10

10 PROGRAMS.

PROGRAMS2 £10

10 PROGRAMS.

PROGRAMS3 £20

25 PROGRAMS.

0 LEVEL REVISION COURSES IN THE FOLLOWING SUBJECTS-£10 EACH

PHYSICS, CHEMISTRY, BIOLOGY, MATHS, FRENCH, GEOGRAPHY, HISTORY.

MICRODRIVE CARTRIDGE INCLUDED
OVERSEA CUSTOMERS ADD £1 FOR POSTAGE.

CHEQUES/POSTAL ORDERS TO-
NEW HORIZONS SOFTWARE,
FOUR WINDS, CUM LANE, ROGERSTONE,
NEWPORT, GWENT, NP1 9AF.

IF YOU HAVE ANY PROGRAMS OF A SUITABLE STANDARD THEN SEND THEM TO THE ABOVE ADDRESS FOR CONSIDERATION FOR MARKETING. TOP ROYALTIES PAID!

FURTHER DETAILS AVAILABLE ON REQUEST.

ENQUIRIES WELCOME (S.A.E. PLEASE)